

Design and Functional Evaluation of a Gui-Based Expense Tracker System Using Python, Tkinter, and MYSQL

Arceso, Rhiane Biatrice P., Bazarte, Jade Monique L., Cagadas, Fresly Ann T.,
Dacara, Lance Louis B., Reyes, Gabrielle B., Sibug, Jhanna Paula R., Engr. Fabro, Meshelle N.

Department of Computer Engineering Eulogio “Amang” Rodriguez Institute of Science and
Technology (EARIST) Nagtahan, Manila, Philippines

DOI: <https://doi.org/10.51583/IJLTEMAS.2026.150500271>

Received: 07 June 2026; Accepted: 12 June 2026; Published: 25 June 2026

ABSTRACT

Manual expense tracking often results in disorganized records, inaccurate calculations, and difficulty in monitoring financial activities. To address these problems, this study developed a GUI-based Expense Tracker System designed to help users record, organize, update, delete, and monitor expenses efficiently. The system was developed using Python as the main programming language, Tkinter for the graphical user interface, and MySQL as the database management system for permanent record storage. The application includes essential expense management features such as Add Expense, View Expenses, Edit or Update Record, Delete Record, Clear All Records, and Report or Analytics Window. It also provides category filtering, automatic summary computation, database connectivity, and light/dark mode interface options to improve usability. Functional testing was conducted to determine whether the system performs correctly under valid and invalid input conditions. The testing covered major operations such as adding records, validating empty fields, detecting non-numeric amounts, checking date formats, filtering categories, updating records, deleting records, generating reports, and verifying database connectivity. The results showed that the system successfully performed all tested functions and maintained accurate synchronization between the graphical interface and the MySQL database. Overall, the Expense Tracker System provides a simple, organized, and user-friendly solution for managing expense records while demonstrating the practical application of object-oriented programming, GUI development, and database integration.

Keywords: Expense Tracker, Python, Tkinter, MySQL, Graphical User Interface, Expense Management, Database Integration

INTRODUCTION

Background of the Study

Managing expenses is an important part of personal and organizational financial management. However, many people still rely on manual methods such as notebooks, paper records, or simple spreadsheets to record their daily spending. These methods may be useful for basic tracking, but they can become disorganized and difficult to manage over time. Manual expense tracking may lead to missing records, inaccurate calculations, duplicated entries, and difficulty in reviewing past transactions. As a result, users may find it challenging to monitor their spending habits and make informed financial decisions.

With the continuous development of technology, computerized systems have become useful tools for improving record management and reducing manual errors. A computerized expense tracking system can help users store financial records in one organized platform, retrieve previous transactions easily, and compute summaries automatically. Through automation, users no longer need to manually calculate totals or search through scattered records. This improves accuracy, saves time, and supports better financial monitoring.

This study focuses on the development of a GUI-based Expense Tracker System using Python, Tkinter, and MySQL. The system is designed to help users record, organize, edit, delete, view, and monitor expense records

through a simple and user-friendly graphical interface. Instead of relying on command-line instructions, users can interact with the system through buttons, tables, forms, menus, and input fields. This makes the system easier to understand and more convenient for users.

The system also uses MySQL as its database management system to provide permanent storage for expense records. Through database integration, records remain saved even after the application is closed. This allows users to access previous expense records, update existing transactions, delete unnecessary entries, and generate summaries whenever needed. The system also includes features such as category filtering, report generation, automatic expense summaries, and light/dark mode to improve usability and user experience.

Overall, the Expense Tracker System was developed to provide a more organized, accurate, and efficient method of managing expense records. It also serves as a practical application of object-oriented programming concepts, graphical user interface development, input validation, error handling, and database management. Through this system, users can monitor expenses more effectively while the developers gain hands-on experience in creating a functional software application.

Statement of the Problem

This study aims to develop a GUI-based Expense Tracker System that can efficiently manage, organize, and monitor expense records.

Specifically, this study seeks to answer the following questions:

1. How can the system automate expense tracking and record management?
2. How can the system store and retrieve expense records efficiently and accurately?
3. How can the system reduce errors commonly encountered in manual expense recording and computation?
4. How can a graphical user interface improve user interaction and usability of the system?

Objectives of the Study

General Objective

The general objective of this study is to design and develop a GUI-based Expense Tracker System using Python, Tkinter, and MySQL that allows users to record, manage, organize, and monitor expense records efficiently.

Specific Objectives

Specifically, this study aims to:

1. Design the system flow and process using a flowchart and structured system design.
2. Develop the system using Python programming language.
3. Create a graphical user interface using Tkinter to provide an easy and user-friendly system.
4. Integrate MySQL database for permanent storage and management of expense records.
5. Implement essential expense management functions such as adding, viewing, updating, deleting, filtering, and generating reports.
6. Apply input validation and error handling to reduce inaccurate or incomplete records.

7. Test the functionality and usability of the system through functional testing.

Significance of the Study

This study is significant because it provides a computerized solution for organizing and monitoring expense records. The Expense Tracker System can help reduce manual work, improve accuracy, and make financial record management easier and more efficient.

Users. The system can help users record and monitor expenses in a faster and more organized way. Through its graphical interface, users can easily add, view, update, delete, and review their expenses without relying on manual records.

Organizations. Small organizations or groups may benefit from the system because it can help them manage simple expense records, monitor spending, and reduce the effort needed for manual tracking and computation.

Students. The study can help students understand the practical application of Python programming, object-oriented programming, graphical user interface development, database integration, and system testing.

Future Developers and Researchers. This study may serve as a reference for future developers who want to improve or create similar systems related to expense tracking, budgeting, financial monitoring, or database-connected desktop applications.

Scope and Limitations

This study focuses on the development of a desktop-based Expense Tracker System using Python, Tkinter, and MySQL. The system includes basic expense management functions such as Add Expense, View Expenses, Update Record, Delete Record, Clear All Records, and Report Generation. It also includes category filtering, automatic expense summaries, database connectivity, input validation, and light/dark mode interface options.

The system is designed for offline desktop use and is mainly intended for learning and academic purposes. It does not yet include advanced features such as user login authentication, multi-user access, cloud storage, mobile application support, automatic online backup, or advanced financial forecasting. The system is limited to managing expense records stored in the connected MySQL database. Future improvements may expand the system by adding stronger security, multi-user support, export options, and more advanced financial analysis tools.

REVIEW OF RELEVANT THEORY, STUDIES, AND LITERATURE

This chapter presents the relevant theories, studies, and literature that support the development of the GUI-Based Expense Tracker System. It discusses computerized management systems, database management systems, Python programming, graphical user interface applications, object-oriented programming, automation, financial management systems, data processing, record management, human-computer interaction, input validation, and related systems. These concepts provide the foundation for understanding the importance of developing a computerized system for recording, organizing, monitoring, and managing expense records.

Computerized Management Systems

Computerized management systems are widely used in handling records, processing data, and improving operational efficiency. Unlike manual record-keeping, computerized systems provide faster data entry, easier access to information, and reduced human error. In an Expense Tracker System, a computerized system allows users to record expenses digitally, monitor balances, categorize transactions, and review previous records more efficiently. This improves accuracy in managing financial information and helps users become more aware of their spending habits.

Manual expense tracking often depends on notebooks, paper records, or spreadsheets. Although these methods can be useful for simple recording, they may become difficult to manage when the number of transactions increases. Records may be misplaced, calculations may be incorrect, and users may find it time-consuming to search for previous expenses. A computerized expense tracker addresses these problems by placing all records in one organized system where expenses can be added, viewed, updated, deleted, filtered, and summarized.

According to Laudon and Laudon (2012), computerized systems improve organizational performance by making data easier to manage, process, and retrieve. Similarly, O'Brien and Marakas (2011) explained that computerized information systems improve decision-making and operational productivity by reducing delays in data processing. These concepts support the development of the Expense Tracker System because the system focuses on improving the way users handle financial records through automation and organized data management.

Database Management Systems

Database Management Systems, or DBMS, are essential in storing and organizing data in an efficient and structured way. A database allows records to be stored permanently and retrieved whenever needed. In an Expense Tracker System, the database stores important transaction details such as expense ID, date, category, amount, and notes. It also allows users to add, update, delete, search, and view expense records accurately.

For this study, MySQL was used as the database management system. MySQL is a relational database management system that allows data to be stored in tables. This makes it suitable for managing expense records because each transaction can be stored as a separate record with specific fields. Through MySQL, the system can maintain organized data and ensure that information remains available even after the application is closed.

The use of a database improves the reliability of the Expense Tracker System. Without a database, the program would only store data temporarily while the application is running. Once the program is closed, unsaved data may be lost. By using MySQL, expense records are stored permanently, making the system more practical and dependable. The database also supports data consistency, faster retrieval, and easier record management.

According to Silberschatz, Korth, and Sudarshan (2019), database systems improve data consistency, security, and accessibility, making them essential in information management applications. Their discussion supports the use of MySQL in this system because accurate and organized data storage is necessary for financial tracking applications.

Python Programming in System Development

Python is a widely used programming language in system development because of its simplicity, readability, and flexibility. It is commonly used in creating desktop applications, automation tools, financial systems, and database-connected programs. Python is also beginner-friendly, making it appropriate for academic projects that require both functionality and understandable code structure.

In the Expense Tracker System, Python was used as the main programming language. It was responsible for processing user inputs, validating records, performing calculations, connecting to the database, and controlling the overall flow of the application. Python also supports libraries such as Tkinter, which was used to create the graphical user interface of the system.

Python is useful in an expense tracker because it can process calculations such as total expenses, today's expenses, monthly expenses, and category-based summaries. It can also perform database operations through SQL commands such as INSERT, SELECT, UPDATE, and DELETE. These commands allow the system to save new records, retrieve stored data, update existing entries, and remove unnecessary records.

According to Bhavani et al. (2025), Python-based expense tracker systems provide efficient transaction recording, automated calculations, and improved financial monitoring. Their findings show that Python is

effective for expense tracking applications because it supports GUI design, database integration, and automated data processing.

Object-Oriented Programming in System Development

Object-Oriented Programming, or OOP, is a programming approach that organizes code into objects and classes. It allows developers to group related data and functions together, making the program easier to understand, maintain, and improve. Since this project was developed under the Object-Oriented Programming course, the use of OOP concepts is important in understanding how the system was structured and developed.

In the Expense Tracker System, OOP principles can help organize the system into separate components. For example, the interface, database connection, expense records, validation functions, and report generation can be structured into different classes or functions. This improves code organization because each part of the system has a specific responsibility.

OOP also supports reusability and maintainability. Instead of writing repeated code, developers can create reusable methods for common tasks such as validating amount input, checking date format, refreshing the dashboard, connecting to the database, and displaying records. This makes the system easier to debug and upgrade in the future.

The use of OOP in the Expense Tracker System supports better program structure and helps the developers apply real-world software development practices. It also strengthens the connection between classroom programming concepts and practical system development.

Graphical User Interface-Based Applications

Graphical User Interface, or GUI, applications make systems easier to use by allowing users to interact through visual elements such as buttons, text fields, tables, menus, tabs, and dialog boxes. Compared with command-line systems, GUI-based applications improve user experience because users do not need to memorize commands. Instead, they can perform tasks by clicking buttons and filling out forms.

In the Expense Tracker System, the GUI was developed using Tkinter. The interface includes major sections such as Dashboard, Add Expense, View Expenses, Edit/Delete, and Report. These sections allow users to manage expense records in an organized and user-friendly environment. The system also includes light and dark mode options, making the interface more adaptable to user preferences.

The GUI improves usability because users can easily add expenses, view records, update transactions, delete entries, filter by category, and generate reports. It also reduces confusion because each function is placed in a specific tab or section. For example, the Add Expense tab is used for creating records, while the Edit/Delete tab is used for modifying or removing records.

Shneiderman et al. (2016) emphasized that effective user interface design improves usability by making systems easier to learn, understand, and operate. This supports the use of Tkinter in the Expense Tracker System because the system aims to provide an accessible and convenient experience for users.

Automation in Information Systems

Automation in information systems helps reduce manual work by allowing tasks to be processed automatically. In an Expense Tracker System, automation is useful because it allows the system to calculate totals, update summaries, categorize expenses, refresh records, and generate reports without requiring the user to compute manually.

Manual expense tracking can be time-consuming because users need to calculate totals by themselves. This may lead to mistakes, especially when there are many expense entries. With automation, the system can instantly compute total expenses, today's expenses, and monthly expenses. This helps users monitor their spending more accurately and conveniently.

Automation also improves decision-making because users can immediately view updated financial information. When a new expense is added, the dashboard automatically reflects the updated total. When a record is deleted or edited, the system refreshes the displayed data. This makes the system more responsive and efficient.

According to Birajdar (2025), automated expense tracking systems improve personal financial management by providing instant expense summaries, reducing calculation errors, and helping users analyze spending behavior. This supports the development of the Expense Tracker System because automation is one of its main functions.

Financial Management Systems

Financial Management Systems are designed to help individuals and organizations monitor income, expenses, budgets, and financial transactions. These systems provide accurate financial records and assist users in making informed financial decisions. Through digital tools, users can analyze spending patterns, control unnecessary expenses, and improve overall financial planning.

In the Expense Tracker System, financial management is supported through expense recording, category classification, summary reports, and analytics. Users can identify where their money is being spent, how much they have spent for the day, and how much they have spent for the month. These features help users become more aware of their spending habits.

Financial management systems are important not only for businesses but also for students, families, workers, and individuals who want to control their budget. A simple expense tracker can help users develop discipline in spending and improve their financial awareness. By reviewing records regularly, users can identify unnecessary expenses and make better decisions.

Bourgeois, Smith, Wang, and Mortati (2019) stated that information systems help manage personal and organizational data by improving accessibility, accuracy, and efficiency of information processing. This supports the development of the Expense Tracker System because it focuses on organizing and managing financial data effectively.

Data Processing and Record Management

Data processing is an important component of information systems because it converts raw data into useful information. In expense tracking applications, raw data includes the amount, date, category, and notes entered by the user. The system processes this data by validating it, storing it in the database, displaying it in tables, and summarizing it through reports.

Record management ensures that financial information remains organized, accurate, and accessible. In an Expense Tracker System, proper record management allows users to review transaction history, search for specific records, update incorrect entries, delete unnecessary records, and generate summaries. These functions are important because users rely on accurate data to monitor their expenses.

Effective record management improves the reliability of the system. If expense records are not properly organized, users may have difficulty understanding their financial activities. By using tables, filters, and reports, the system helps users review their expenses more clearly and efficiently.

Bourgeois et al. (2019) emphasized that properly managed information systems improve information quality, operational efficiency, and decision-making processes. Their findings support the use of organized record management in financial tracking applications.

Input Validation and Error Handling

Input validation and error handling are important in system development because they prevent incorrect data from being saved and help users correct mistakes. In an Expense Tracker System, possible errors include

empty fields, non-numeric amount input, invalid date format, no selected record for editing or deleting, and incorrect user actions.

The system uses validation to check whether the required fields are complete before saving a record. It also checks whether the amount entered is numeric and whether the date follows the correct format. If the user enters invalid information, the system displays an error message and prevents the record from being saved. This improves data accuracy and prevents database errors.

Error handling also improves user experience because it guides users when something is wrong. Instead of crashing, the system provides clear messages such as asking the user to complete required fields or enter a valid amount. This makes the system more reliable and easier to use.

In this study, input validation and error handling are important because they support data integrity. They ensure that only valid and complete expense records are stored in the MySQL database. These features also help protect the system from common user mistakes.

Human-Computer Interaction in GUI Systems

Human-Computer Interaction, or HCI, focuses on how users interact with computer systems. It emphasizes usability, accessibility, efficiency, and user satisfaction. A system with good HCI design allows users to complete tasks easily and with fewer errors.

In financial applications such as Expense Tracker Systems, HCI is important because users need to input, review, and manage records regularly. If the interface is confusing, users may make errors or avoid using the system. A well-designed interface helps users understand the available features and complete tasks efficiently.

The Expense Tracker System supports HCI principles by using organized tabs, readable labels, buttons, tables, confirmation dialogs, and theme options. The Dashboard gives users a quick overview of their expenses, while the Add Expense, View Expenses, Edit/Delete, and Report tabs separate the functions clearly. This organization makes the system easier to navigate.

According to Shneiderman et al. (2016), effective user interfaces improve usability by making systems easier to learn, understand, and operate. These principles are relevant to the Expense Tracker System because the system relies on its GUI to provide a user-friendly experience.

Related Systems

Several existing systems are related to the proposed Expense Tracker System. Examples include Microsoft Excel, Mint, Wallet, Money Manager, and other budgeting or financial tracking applications. These systems allow users to record expenses, create budgets, review spending, and generate reports.

Microsoft Excel can be used for expense tracking through tables and formulas. It allows users to calculate totals and organize financial records manually or semi-automatically. However, it may still require spreadsheet knowledge and manual formatting. Compared with Excel, the Expense Tracker System provides a more direct interface where users can add, view, update, and delete records through buttons and forms.

Mint is a financial management application that allows users to monitor expenses, create budgets, and track financial activities. Wallet provides expense recording and budget planning features, while Money Manager is commonly used for tracking daily expenses and generating financial reports. These systems are related to the proposed Expense Tracker because they share similar functions such as transaction recording, category organization, and expense monitoring.

Although these existing systems may offer more advanced features, the proposed Expense Tracker System focuses on the basic and essential functions needed for simple financial record management. It is designed mainly for academic and learning purposes, showing how Python, Tkinter, MySQL, and OOP concepts can be applied in developing a functional desktop application.

Synthesis of Reviewed Literature and Studies

The reviewed literature and studies show that computerized systems, database management, Python programming, graphical user interface design, automation, and record management are important components in developing an effective Expense Tracker System. Computerized systems improve the speed and accuracy of record handling, while database systems provide permanent and organized storage. Python supports system development through its simplicity and flexibility, while Tkinter provides the graphical interface needed for user interaction.

The literature also shows that automation improves financial monitoring by reducing manual computation and providing instant summaries. Input validation and error handling strengthen the reliability of the system by preventing incorrect data from being saved. Human-computer interaction principles support the importance of designing an interface that is easy to understand and operate.

Overall, the related theories, studies, and literature support the development of the GUI-Based Expense Tracker System. The system addresses the problems of manual expense tracking by providing a computerized, organized, and user-friendly application that allows users to record, manage, and monitor expenses more efficiently.

Conceptual Framework

The conceptual framework of this study follows the Input-Process-Output model.

The input includes expense information such as amount, category, date, and notes. These data are entered by the user through the graphical user interface.

The process includes input validation, database storage, record retrieval, record updating, deletion, category filtering, summary computation, and report generation. The system checks whether the entered data are complete and valid before saving them to the MySQL database. It also processes stored records to generate summaries and reports.

The output includes organized expense records, updated transaction lists, category-filtered results, expense summaries, analytics reports, and accurate total expense computations. These outputs help users monitor and manage their financial records more effectively.

Through this framework, the Expense Tracker System transforms raw expense data into organized and useful financial information.

METHODOLOGY

This chapter presents the research design, system development process, system design, database design, tools and technologies used, system features, testing procedure, and evaluation method used in the development of the GUI-Based Expense Tracker System. It explains how the system was planned, designed, developed, tested, and implemented to provide users with an organized and efficient way of managing expense records.

Research Design

This study used a system development approach. This design was appropriate because the main purpose of the study was to develop a functional desktop-based Expense Tracker System that can record, organize, monitor, update, delete, and summarize expense records through a graphical user interface.

The system was developed to address common problems in manual expense tracking, such as disorganized records, inaccurate calculations, difficulty in retrieving past transactions, and inefficient financial monitoring. Through the system development approach, the researchers were able to identify the problem, design the system flow, create the graphical interface, integrate a database, test the system functions, and implement the final application.

The Expense Tracker System was developed using Python as the main programming language, Tkinter as the graphical user interface framework, and MySQL as the database management system. These tools were selected because they support desktop application development, user-friendly interface design, and permanent record storage.

System Development Process

The development of the Expense Tracker System followed several stages to ensure that the application would be functional, accurate, and user-friendly.

Planning

During the planning stage, the researchers identified the main problem and purpose of the system. The researchers observed that manual expense tracking can be time-consuming, inaccurate, and difficult to organize. Based on this problem, the researchers planned a computerized system that would allow users to record, view, edit, delete, filter, and summarize expenses more efficiently.

The researchers also identified the required system features, including Add Expense, View Expenses, Edit or Update Record, Delete Record, Clear All Records, Report or Analytics Window, category filtering, and light/dark mode. These features were included to make the system useful, organized, and convenient for users.

System Design

The system design stage involved creating the system flowchart, database structure, and graphical user interface layout. The flowchart served as a guide for the overall operation of the system. It showed the process of opening the application, connecting to the database, selecting a function, validating user input, saving records, updating records, deleting records, generating reports, and exiting the system.

The graphical user interface was also planned during this stage. The system was designed with navigation tabs such as Dashboard, Add Expense, View Expenses, Edit/Delete, and Report. These sections were arranged to make the system easier to use and understand.

Development

The development stage involved the actual coding and implementation of the system. Python was used to create the logic and functionality of the program. Tkinter was used to design the graphical user interface, including buttons, labels, entry fields, dropdown menus, tables, tabs, and dialog boxes.

MySQL was integrated into the system to store and manage expense records permanently. Through database integration, records remain saved even after the application is closed. The system was programmed to perform important database operations such as inserting new records, retrieving records, updating existing records, deleting selected entries, and clearing all stored records when confirmed by the user.

Object-Oriented Programming principles were also applied to organize the program structure. Functions and system components were arranged properly to improve readability, maintainability, and reusability of the code.

Testing

After the system was developed, it underwent functional testing. The researchers tested each feature to determine whether the system produced the expected output. Both valid and invalid inputs were tested to check the accuracy of input validation, error handling, database connectivity, and system response.

The testing process included adding expense records, viewing records, filtering by category, updating records, deleting records, clearing all records, generating reports, switching between light and dark mode, using keyboard shortcuts, checking dashboard summaries, and verifying database connectivity.

Implementation

After successful testing and debugging, the final version of the Expense Tracker System was implemented as a desktop-based application. The completed system provided users with an interactive platform for monitoring and managing expenses. It allowed users to handle expense records more efficiently compared to manual recording methods.

System Design

The Expense Tracker System begins by launching the application dashboard and connecting to the MySQL database. Once the application is opened, the dashboard displays summary information such as total expenses, today's expenses, and monthly expenses. It also shows a detailed summary table containing stored expense records.

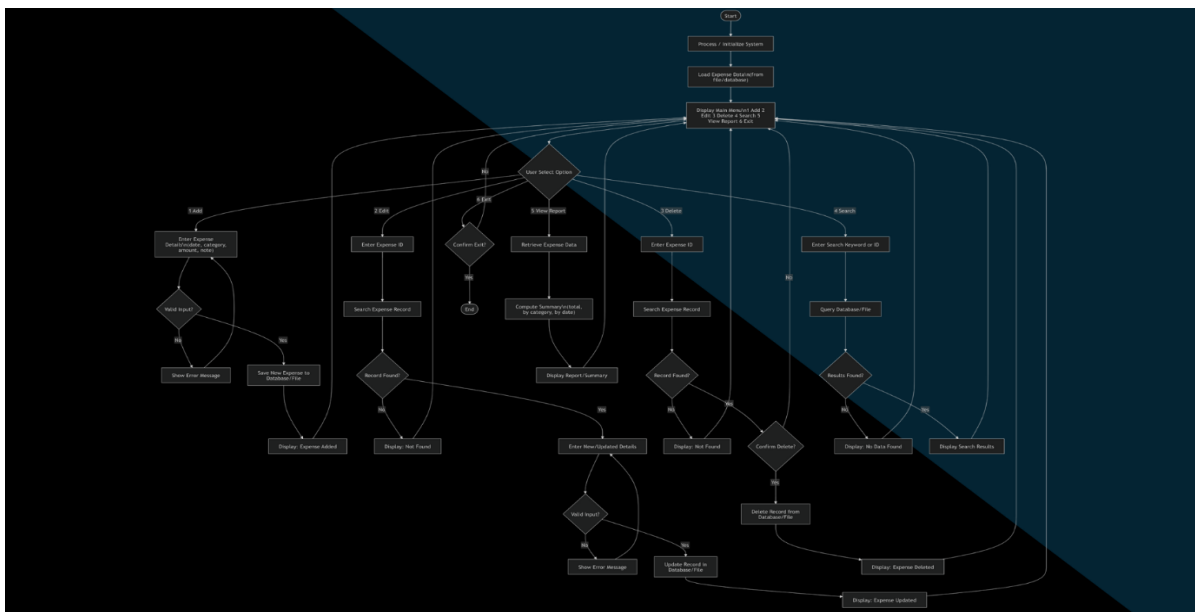


Figure 1. Expense Tracker System Flowchart

This figure presents the overall process of the Expense Tracker System, including the movement from the dashboard to different system functions such as adding, viewing, editing, deleting, clearing, and generating expense reports.

The system contains several navigation tabs. The Dashboard tab displays the expense overview. The Add Expense tab allows the user to enter new expense details. The View Expenses tab displays stored records and allows filtering by category. The Edit/Delete tab allows the user to update or remove selected records. The Report tab opens the analytics report window, which presents financial summaries and expense details.

When the user adds a new expense, the system asks for the amount, category, date, and optional notes. The entered information is validated before it is saved into the database. If the input is incomplete or invalid, the system displays an error message. If the input is valid, the record is saved in the MySQL database and the dashboard is refreshed.

When the user edits a record, the selected expense information is displayed in the input fields. The user can modify the amount, category, date, or notes. After saving, the system validates the updated information and updates the database.

When the user deletes a record, the selected expense is removed from the database. The table and dashboard totals are then updated automatically. The Clear All function allows users to remove all records, but the system first displays a confirmation message to prevent accidental deletion.

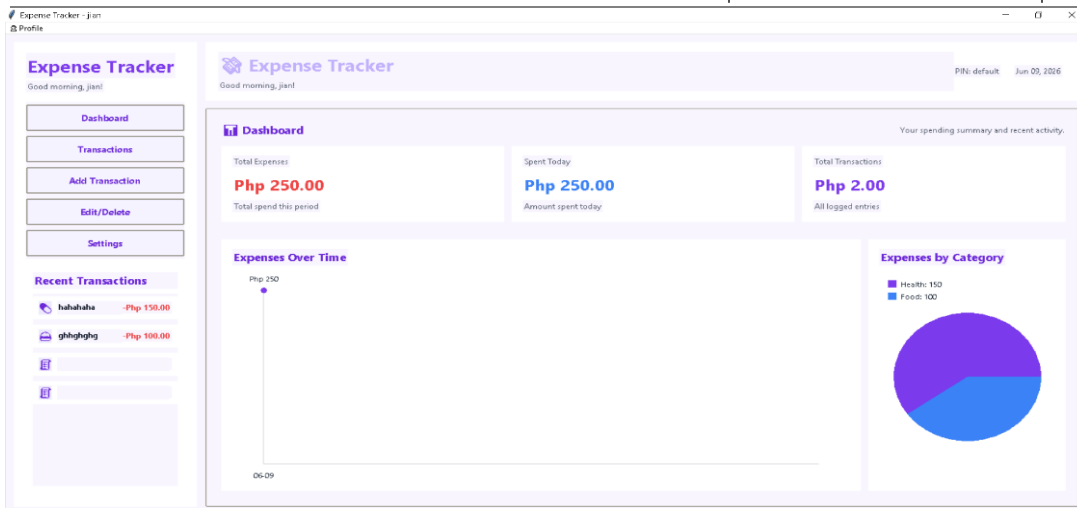


Figure 2. Expense Tracker System Dashboard

This figure shows the main dashboard of the system, where users can view total expenses, today's expenses, monthly expenses, and the detailed summary of recorded transactions.

Database Design

The Expense Tracker System uses MySQL as its database management system. The database stores expense records in an organized and permanent manner. Each expense record includes important details such as expense ID, date, category, amount, and notes.

The database supports the main operations of the system: creating, reading, updating, and deleting records. It allows the system to save new expense entries, display existing records, update selected transactions, and remove records when needed.

The use of MySQL improves the reliability of the system because records are stored permanently and can be retrieved whenever the application is opened. It also helps maintain data consistency between the graphical user interface and the database.

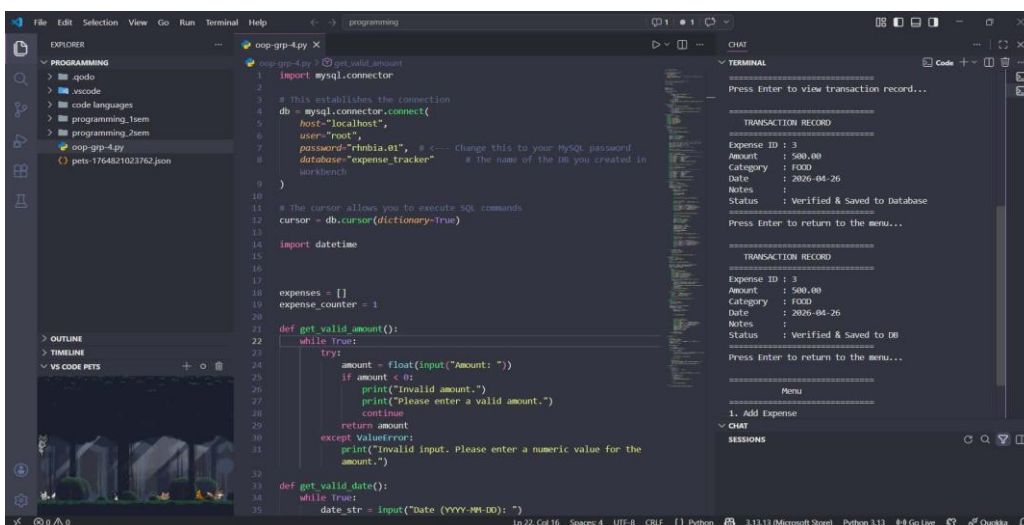


Figure 3. MySQL Database Integration of the Expense Tracker System

This figure shows the database connection of the Expense Tracker System using MySQL in Python. It also presents the transaction record displayed in the terminal, showing that the expense entry was successfully verified and saved to the database.

Tools and Technologies Used

The following tools and technologies were used in developing the Expense Tracker System:

Python was used as the main programming language. It handled the system logic, calculations, input validation, database operations, and program flow.

Tkinter was used to create the graphical user interface. It provided the visual components of the system, such as buttons, entry fields, tabs, tables, menus, and dialog boxes.

MySQL was used as the database management system. It stored expense records permanently and allowed the system to retrieve, update, and delete data efficiently.

VS Code was used as the development environment for writing, editing, debugging, and testing the program code.

System Features

The Expense Tracker System includes the following features:

Add Expense allows users to record new expenses by entering the amount, category, date, and notes.

View Expenses displays all saved expense records in a table and allows users to filter records by category.

Update Record allows users to edit selected expense records and save the updated information in the database.

Delete Record allows users to remove selected expense records from the database.

Clear All Records allows users to delete all stored records after confirming the action.

Report or Analytics Window displays expense summaries, including total expenses, monthly expenses, total record count, and highest-spend category.

Light/Dark Mode allows users to switch between light and dark themes based on their preference.

Input Validation and Error Handling prevent invalid data such as empty fields, non-numeric amounts, and incorrect date formats from being saved.

Testing Procedure

The system was tested using functional testing. The researchers manually tested the system by simulating real user actions. The purpose of the testing was to determine whether each feature worked correctly and whether the system could handle both valid and invalid inputs.

The following test areas were included:

Test Area	Purpose
Add Expense	To check if valid expense records can be saved successfully.
Empty Field Validation	To check if the system detects missing required fields.
Invalid Amount Validation	To check if the system rejects non-numeric amount input.
Invalid Date Validation	To check if the system detects incorrect date format.
View Expenses	To check if stored records are displayed properly.

Category Filtering	To check if expenses can be filtered based on category.
Update Record	To check if selected records can be edited and saved.
Delete Record	To check if selected records can be removed from the database.
Clear All Records	To check if all records can be deleted after confirmation.
Report Generation	To check if the analytics report displays correct summaries.
Dark/Light Mode	To check if the interface theme changes correctly.
Database Connectivity	To check if the system connects properly to the MySQL database.
Dashboard Summary	To check if total expenses, daily expenses, and monthly expenses are computed accurately.

The testing confirmed whether the system could perform its intended functions and respond properly to user actions.

Evaluation of the System

The Expense Tracker System was evaluated based on functionality, accuracy, usability, reliability, and database performance.

Functionality was evaluated by checking whether the main features of the system worked properly, including adding, viewing, updating, deleting, clearing, filtering, and generating reports.

Accuracy was evaluated by checking whether the system computed expense summaries correctly and stored records accurately in the database.

Usability was evaluated based on the clarity and organization of the graphical user interface. The tab-based layout, buttons, tables, dropdown menus, and dialog boxes helped make the system easier to navigate.

Reliability was evaluated by checking whether the system consistently saved, retrieved, updated, and deleted records without errors.

Database performance was evaluated by checking whether the system maintained proper synchronization between the Tkinter interface and the MySQL database.

Summary

The methodology of this study followed a system development process consisting of planning, system design, development, testing, and implementation. The Expense Tracker System was developed using Python, Tkinter, MySQL, and VS Code. The system was designed to provide users with an organized and efficient way to manage expense records through a desktop-based graphical user interface.

Through this methodology, the researchers were able to develop a functional Expense Tracker System that supports expense recording, viewing, updating, deleting, clearing, filtering, report generation, theme switching, input validation, and database storage. The system demonstrates the practical application of object-oriented programming, graphical user interface development, database integration, and functional testing.

RESULTS AND DISCUSSION

This chapter presents the results and discussion of the developed GUI-Based Expense Tracker System. It discusses the system overview, completed system features, graphical user interface, testing results, and overall performance of the system. The results show how the Expense Tracker System was able to record, organize, update, delete, filter, summarize, and manage expense records using Python, Tkinter, and MySQL.

System Overview

The Expense Tracker system is a desktop-based application designed to help users efficiently record, monitor, and manage their personal or organizational expenses. The system addresses the need for an organized, accurate, and accessible means of tracking financial outflows without relying on manual logs or spreadsheets.

The system operates through a menu-driven graphical user interface (GUI) built with Python's Tkinter library and integrates with a MySQL relational database for persistent data storage. Upon launching the application, the system automatically loads all existing expense records from the database, ensuring that users have immediate access to their financial data without additional setup steps.

The primary goal of the system is to improve the accuracy, organization, and efficiency of expense tracking. It achieves this through input validation that prevents erroneous entries, by category classification of expenses, and summary reporting that gives users an at a glance view of their spending. The system also supports both light and dark display themes, making it adaptable to different user preferences and working environments.

Overall, the Expense Tracker system provides a complete lifecycle for expense records: from entry and viewing, to editing, deletion, and report generation within a single, cohesive interface.

System Features

The Expense Tracker system is equipped with six core features that collectively support a complete expense management workflow. Each feature is designed with usability and data integrity in mind, incorporating input validation and confirmation mechanisms to prevent accidental errors.

Add Record

The Add Expense feature allows users to log new expense entries into the system. To add a record, the user fills in the required fields: Amount (in Philippine Peso), Category, and Date. An optional Notes field is also available for additional context. The category selection is presented as a dropdown menu with predefined options including Food, Transportation, Shopping, Health, and Others, with the option to specify a custom category under Others (type). The date field is pre-populated with the current date and supports a calendar picker for easy selection.

Once all required fields are completed, the user clicks the 'Add Expense' button to save the entry. The system validates all inputs before saving. If any required field is empty or if an invalid value (such as non-numeric text in the Amount field or an incorrectly formatted date) is detected, an error dialog is displayed and the record is not saved. Upon successful validation, the record is written to the MySQL database and the dashboard is refreshed to reflect the updated totals.

View Records

The View Expenses feature presents all recorded expenses in a tabular list, displaying each record's ID, Date, Category, and Amount. Users may filter the displayed records by category using a dropdown filter at the top of the view, allowing them to quickly locate expenses belonging to a specific category such as Food or Transportation. Selecting any record from the list and clicking the 'Open Selection' button opens an Expense Details popup, which presents a formatted receipt showing the Receipt ID, Date, Category, Amount, and any associated Notes.

Update Record

The Edit or Update Record Expense feature provides users with the ability to modify existing expense records. To Update or Edit a record, the user navigates to the Edit or Update tab and selects the target record from the displayed table. The record's current details Amount, Category, Date, and Notes are automatically populated into the editable fields at the bottom of the interface. The user may then modify any of the fields and click the 'Save' button to apply the changes. The system validates the updated input in the same manner as the Add

Expense feature, ensuring that only valid data is committed to the database. If no record is selected when Save is clicked, the system displays a warning prompting the user to select a record first. Upon a successful save, the database is updated and the expense list is refreshed.

Delete Record

The Delete Record function allows users to permanently remove a selected expense entry from the database. To delete a record, the user selects the desired row in the Edit/Delete tab's table and clicks the 'Delete' button. The record is immediately removed from the database and the expense list is updated accordingly. If no record is selected when Delete is clicked, a warning message is shown instructing the user to select a record before proceeding. This safeguard ensures that records are never accidentally removed due to unintended clicks.

Clear Fields

The Clear All function, accessible from the top toolbar of the main window, allows users to wipe all expense records from the database in a single action. This feature is intended for scenarios where the user wishes to reset the system entirely, such as at the start of a new budget period.

To prevent accidental data loss, the system displays a confirmation dialog before executing the deletion, asking the user to confirm whether they wish to delete all expenses permanently. If the user selects 'Yes,' all records are removed and the dashboard is reset to zero. If the user selects 'No,' the operation is cancelled and all records remain intact.

Exit System

The Exit System function allows users to close the application through two methods: clicking the standard window close button (the 'X' in the top-right corner of the window), or selecting 'Exit' from the File menu in the menu bar. Both methods terminate the application session gracefully, ensuring that any committed database changes are preserved. Keyboard shortcuts are also available for navigation and common actions, as listed in the Help menu's Shortcuts dialog.

System Interface

The Budget/Expense Manager features a modern and clean graphical user interface (GUI) developed using Python's Tkinter library. The interface is organized into tabs for easy navigation and supports both light and dark display themes to accommodate different user preferences. The following subsections describe each major interface screen.

Main GUI Window (Dashboard)

The main window of the application is the Dashboard, which serves as the central hub for the user's expense overview. Upon launching the system, the Dashboard is displayed by default and presents three key summary metrics at a glance: Total Expenses (the cumulative sum of all recorded expenses), Today's Expenses (the total amount spent on the current date), and This Month's Expenses (the total spending within the current calendar month). Below these summary cards, a Detailed Summary section displays all expense records in a table organized by ID, Date, Category, Amount, and Notes.

At the top of the window, the main navigation tab bar provides quick access to the five primary sections of the application: Dashboard, Add Expense, View Expenses, Edit/Delete, and Report. The top-right corner houses the Clear All button and the theme toggle button, which switches the interface between Light Mode and Dark Mode.

The menu bar at the top of the window contains three menus: File (with options for New Expense, Refresh Dashboard, and Exit), Theme (for switching display modes), Tools (for clearing all data), and Help (which provides a Shortcuts reference and an About panel detailing the system's features, version, database, and technology stack).

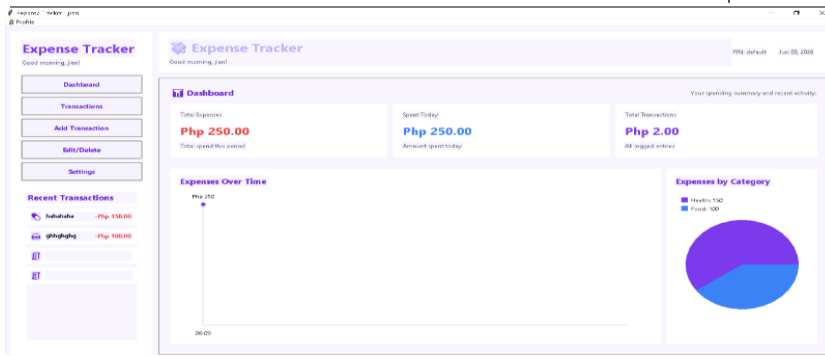


Figure 4. Main GUI Window or Dashboard

The dashboard serves as the main window of the system. It displays the main expense overview, including total expenses, today's expenses, and monthly expenses. It also includes a detailed summary table that shows saved expense records. The dashboard gives users a quick overview of their spending.

Add Expense Interface

The Add Expense tab presents a form with four input fields. The Amount field accepts numerical values representing the cost in Philippine Peso. The Category field is a dropdown with predefined options: Food, Transportation, Shopping, Health, Others, and Others (type) for custom categories. The Date field defaults to the current date in YYYY-MM-DD format and is accompanied by a calendar picker icon that opens a graphical date selection dialog. The Notes field is optional and accepts free-text input for additional context about the expense.

Two action buttons appear at the bottom of the form: Add Expense, which validates and submits the entry, and Clear Fields, which resets all form inputs to their default state. The interface is consistent in both light and dark themes.

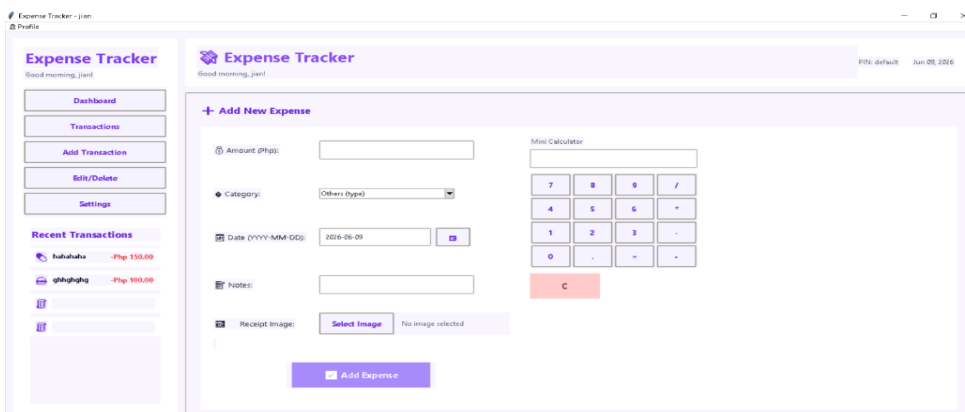


Figure 5. Add Expense Interface

The Add Expense interface allows users to enter new expense records. It includes input fields for amount, category, date, and notes. It also includes buttons for adding the expense and clearing the input fields. This interface provides a structured and simple way of recording new transactions.

View Expenses Interface

The View Expenses tab displays a scrollable list of all expense records in a table with columns for ID, Date, Category, and Amount. A filter dropdown labeled 'Filter by category' at the top allows users to narrow the displayed records to a specific category. To the right of the filter is the 'Open Selection' button, which opens a detailed receipt popup for the currently selected record.

The Expense Details popup presents a formatted digital receipt displaying the Receipt ID, Date, Category,

Amount, and Notes for the selected record.

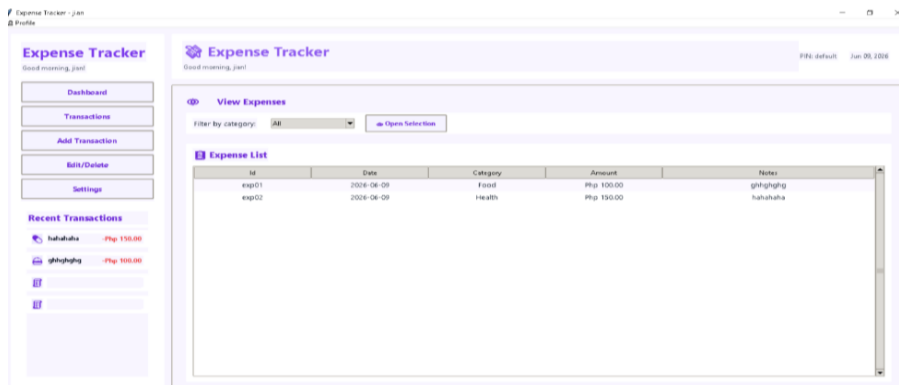


Figure 6. View Expenses Interface

The View Expenses interface displays saved records in a table. It also includes a category filter and an Open Selection button. This helps users browse, filter, and inspect expense records easily.

Update Record Interface

The Edit/Delete tab combines both update and delete functionalities within a single view. The upper portion of the tab displays the full expense table, identical in structure to the View Expenses list. The lower portion contains an editable form that is populated automatically when a record is selected from the table above.

The form includes the same fields as the Add Expense form (Amount, Category, Date, and Notes) and features three action buttons: Save (to commit the updated values to the database), Delete (to remove the selected record), and Clear Fields (to reset the form). This unified layout streamlines the process of record management by keeping both actions accessible from a single screen.

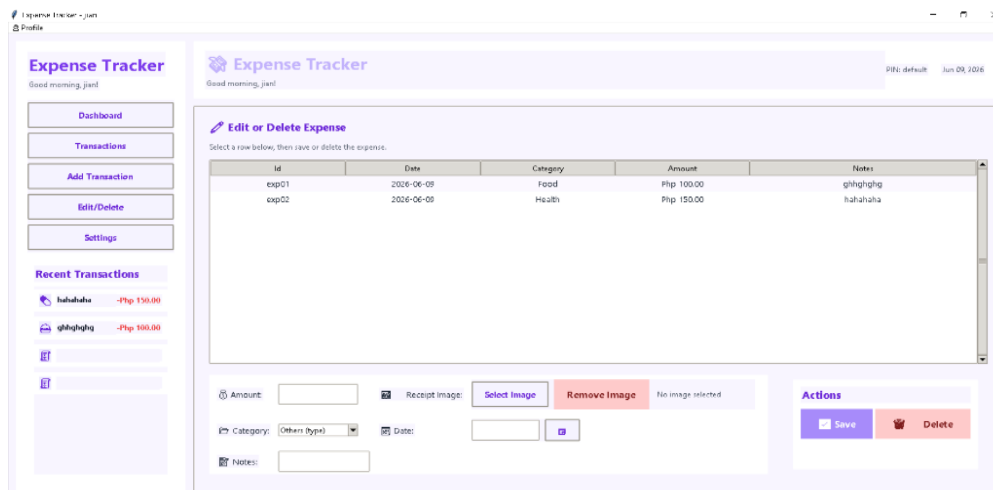


Figure 7. Update Record Interface

The Update Record interface allows users to modify existing expense records. When a record is selected, its details appear in the editable fields. This makes the updating process easier and reduces the chance of editing the wrong information.

Delete Record Interface

As described in the previous subsection, the deletion workflow is integrated within the Edit/Delete tab. The user selects a record from the expense table and clicks the 'Delete' button to remove it. No additional confirmation dialog is shown for individual record deletion; the action is immediate upon clicking the

button and a row is selected. The expense list and dashboard totals are automatically refreshed after deletion

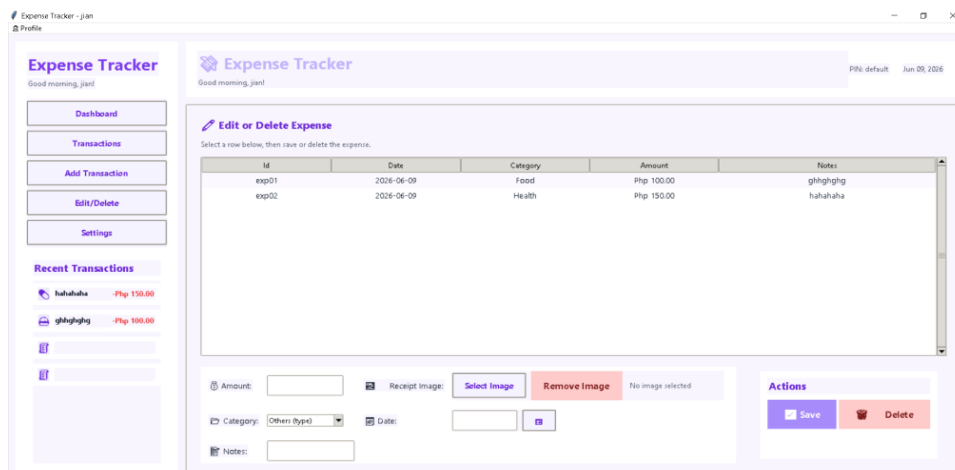


Figure 8. Delete Record Interface

The Delete Record function is included in the Edit/Delete tab. Users can select a record and delete it from the database. The table and dashboard are updated after deletion.

Report Interface

The Report tab provides access to the expense analytics feature. It contains a brief description and a prominent 'Open Report Window' button that launches a dedicated analytics popup window.

The Analytics Report window displays a summary bar showing the total expenses, current month expenses, total record count, and the highest-spend category. Below the summary bar, a scrollable table lists all expense records with their Date, Category, Notes, and Amount. This gives users a consolidated view of all their spending at a glance.

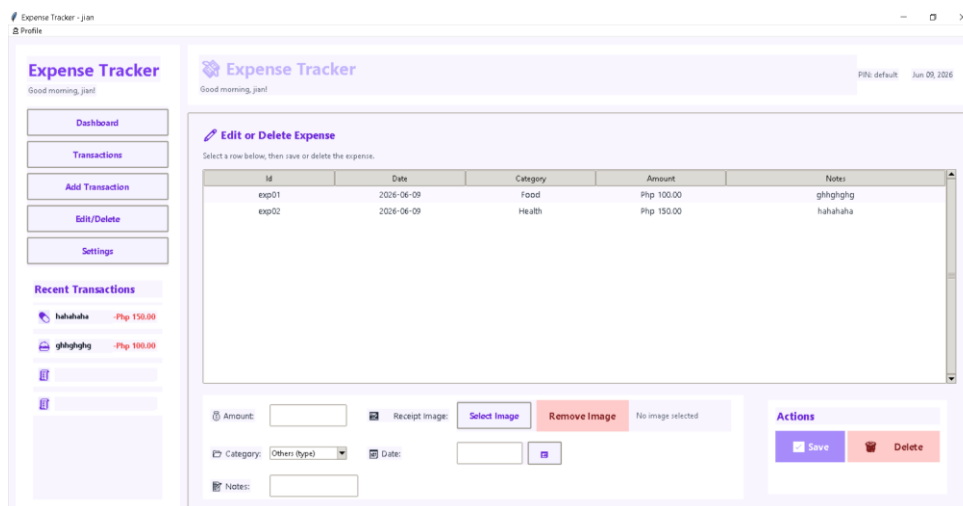


Figure 9. Report Interface

The Report interface allows users to open the analytics report window. The report window displays financial summaries and expense details. This gives users a clearer view of their overall spending.

The use of a graphical user interface improved the usability of the system. Since users can interact with the system through buttons, tabs, tables, dropdown lists, and dialog boxes, the system is easier to operate compared to a command-line program.

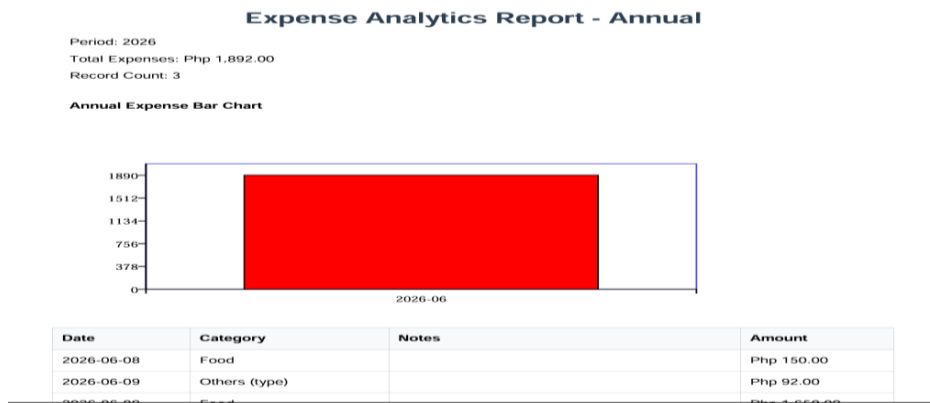


Figure 10. Report Tab Interface

System Testing Results

The Expense Tracker system underwent functional testing to verify that each feature of the system operates correctly under both valid and invalid conditions. Testing was conducted manually by the development team, simulating real user interactions with the application across all major functional areas: record addition, viewing, updating, deletion, report generation, and system-level operations.

The testing approach focused on two categories of test scenarios. First, positive test cases were executed using valid inputs to confirm that the system performs its intended function saving records, displaying data, and producing accurate summaries. Second, negative test cases were run using invalid or incomplete inputs to verify that the system's input validation and error handling mechanisms respond correctly, displaying appropriate error messages without crashing or corrupting data.

Database integration was also tested to ensure that all changes made through the GUI additions, updates, and deletions are accurately reflected in the MySQL database in real time. MySQL Workbench was used as an independent verification tool to query the database and confirm that records match the data displayed in the application interface.

The system underwent functional testing to determine whether each feature worked correctly under valid and invalid conditions. The researchers tested the system manually by simulating actual user actions. Testing covered record addition, viewing, filtering, updating, deleting, clearing records, report generation, theme switching, keyboard shortcuts, database connectivity, and dashboard summary accuracy.

A total of twenty (20) test cases were executed. All test cases produced results that matched the expected outcomes, confirming that the system is functionally stable and that its input validation and error handling mechanisms are effective. The complete test case results are summarized in the table below.

Table 1.

Test Case Number	Test Case	Test Input	Expected Result	Actual Result	Status
TC-01	Add Expense – Valid Input	Amount: ₱500, Category: Food, Date: 2026-05-24	Record saved to database	Record saved successfully; dashboard updated	PASS
TC-02	Add Expense – Empty Amount Field	Amount: blank, Category: Others, Date: 2026-05-24	System prompts user to complete required fields	Error dialog displayed: “Please fill in all required fields”	PASS

TC-03	Add Expense – Invalid Amount (Text)	Amount: “abc”, Category: Food, Date: 2026-05-24	System shows error message for invalid input	Error dialog displayed: “Invalid amount. Please enter a numeric value”	PASS
TC-04	Add Expense – Invalid Date Format	Amount: ₱100, Category: Health, Date: 05-24-2026	System shows date format error	Error dialog displayed: “Invalid date format. Use YYYY-MM-DD”	PASS
TC-05	View Expense Records	Click “View Expenses” tab	All records displayed in the expense list table	All records displayed correctly with ID, Date, Category, and Amount	PASS
TC-06	View Expense – Filter by Category	Select category filter: “Food”	Only Food category records displayed	List filtered to show only Food records	PASS
TC-07	View Expense – Open Receipt	Select record, click “Open Selection”	Receipt popup shows expense details	Receipt dialog displayed with ID, Date, Category, Amount, and Notes	PASS
TC-08	Update Expense – Valid Update	Select exp01, change Amount to ₱200, click “Save”	Record updated in database	Record updated; dashboard totals refreshed	PASS
TC-09	Update Expense – No Record Selected	Click “Save” without selecting a record	System shows prompt to select a record first	Warning dialog: “Please select a record to edit”	PASS
TC-10	Delete Expense – Valid Delete	Select exp02, click “Delete”	Record removed from database	Record deleted; list and dashboard refreshed	PASS
TC-11	Delete Expense – No Record Selected	Click “Delete” without selecting a record	System shows prompt to select a record	Warning dialog: “Please select a record to delete”	PASS
TC-12	Clear All Records	Click “Clear All” button	Confirmation dialog appears before deletion	Confirmation dialog shown; upon confirmation, all records deleted	PASS
TC-13	Clear All – Cancel Confirmation	Click “Clear All,” then click “No”	No records deleted; system remains unchanged	System unchanged; all records retained	PASS
TC-14	Generate Expense Report	Click “Report” tab, then “Open Report Window”	Analytics report window with expense breakdown	Report window displayed with total, monthly, and top category data	PASS
TC-15	Export Receipt as PDF	Open receipt, click “Export/Print”	PDF receipt saved to designated folder	PDF file saved successfully with confirmation message	PASS
TC-16	Dark/Light Mode Toggle	Click “Dark Mode” / “Light Mode” button	UI theme switches between dark and light	Theme switched successfully without data loss	PASS
TC-17	Keyboard Shortcut – Add Expense (Ctrl + N)	Press Ctrl + N	Navigate to Add Expense tab	Add Expense tab activated	PASS
TC-18	Keyboard Shortcut – Refresh Dashboard (F5)	Press F5	Dashboard totals refreshed	Dashboard data refreshed from database	PASS

TC-19	Database Connectivity	Launch application	Application connects to database	Database connected successfully	PASS
TC-20	Dashboard Summary – Totals Accuracy	View dashboard after adding all 6 sample records	Total Expenses = ₱1,486.00	Dashboard displayed ₱1,486.00 correctly	PASS

A total of twenty test cases were conducted, and all test cases passed. This means that the system successfully performed the expected functions and handled error conditions properly.

Test Area	Result
Add Expense with valid input	Record was successfully saved and dashboard was updated.
Empty amount field	Error message was displayed and record was not saved.
Invalid amount input	Error message was displayed for non-numeric input.
Invalid date format	Error message was displayed and record was not saved.
View expense records	All saved records were displayed correctly.
Filter by category	Records were filtered based on the selected category.
Open receipt/details	Selected expense details were displayed correctly.
Update valid record	Record was updated in the database and dashboard was refreshed.
Update without selected record	Warning message was displayed.
Delete valid record	Selected record was deleted and dashboard was refreshed.
Delete without selected record	Warning message was displayed.
Clear all records	Confirmation dialog appeared before deleting all records.
Cancel clear all	Records remained unchanged.
Generate expense report	Analytics report displayed total, monthly, and top category data.
Export receipt as PDF	PDF receipt was saved successfully.
Dark/Light mode toggle	Theme changed successfully without data loss.
Ctrl + N shortcut	Add Expense tab was activated.
F5 shortcut	Dashboard data was refreshed.
Database connectivity	System connected successfully to the database.
Dashboard summary accuracy	Total expenses were displayed correctly.

The test results showed that the Expense Tracker System met its functional requirements. The system was able to correctly process valid inputs and reject invalid inputs. It also maintained proper synchronization between the Tkinter interface and the MySQL database.

DISCUSSION OF FINDINGS

The results of the system development and testing show that the Expense Tracker System successfully achieved its purpose of improving expense record management. The system provided users with a more organized and efficient way to record, monitor, update, delete, and analyze expenses.

The use of MySQL improved the reliability of the system because expense records were stored permanently. Unlike manual records or temporary storage, the database allowed the system to keep records saved even after the application was closed. This made the system more dependable for continuous expense monitoring.

The system also reduced errors commonly encountered in manual tracking. Input validation prevented incomplete records, non-numeric amount entries, and incorrect date formats from being saved. Warning

messages and confirmation dialogs also guided users and helped prevent mistakes, especially when updating, deleting, or clearing records.

Another important result was the successful generation of automated summaries. The dashboard and report window helped users view their total expenses, today's expenses, monthly expenses, record count, and highest-spend category. These summaries can help users understand their spending habits and make better financial decisions.

The graphical user interface also contributed to the success of the system. The tab-based layout made each function easier to locate. Users can add, view, edit, delete, and generate reports without using complicated commands. The light and dark mode option further improved user experience by allowing users to adjust the appearance of the system according to their preference.

Overall, the findings show that the Expense Tracker System is functional, accurate, reliable, and user-friendly within the scope of the study.

Strengths of the System

The system has several strengths based on the results of development and testing.

First, it provides an organized way of managing expense records through a single application. Second, it uses a MySQL database, which allows permanent storage and reliable retrieval of expense records. Third, it includes input validation and error handling, which help prevent inaccurate or incomplete data. Fourth, it provides automated summaries and analytics reports, allowing users to monitor expenses more easily. Fifth, it has a simple graphical user interface that improves usability. Lastly, its light and dark mode feature makes the system more flexible for different user preferences.

These strengths show that the system is useful for basic expense management and can help users improve their financial tracking process.

Limitations Observed

Although the system successfully performed its intended functions, some limitations were observed. The system is designed as a desktop-based application, which means it can only be used on the computer where it is installed. It does not yet support online access, cloud storage, or mobile access.

The system also does not include user login authentication or multi-user support. This means that all users who can access the application may also access the same records. In addition, while the system can generate reports, future versions may improve report customization, automatic report generation, and export options.

The system is also intended mainly for academic and learning purposes. It demonstrates the application of Python, Tkinter, MySQL, and object-oriented programming concepts, but it may still need further improvement before being used as a full-scale financial management system.

Summary

The results showed that the GUI-Based Expense Tracker System successfully performed its intended functions. It allowed users to add, view, update, delete, clear, filter, and summarize expense records. The system also generated reports, switched between light and dark mode, supported keyboard shortcuts, connected properly to the MySQL database, and displayed accurate dashboard summaries.

All twenty functional test cases passed, showing that the system is stable and reliable within its intended scope. The discussion further showed that the system improved the accuracy, organization, and efficiency of expense tracking. Therefore, the Expense Tracker System can be considered a functional and user-friendly tool for managing expense records.

CONCLUSIONS AND RECOMMENDATIONS

Conclusions

Based on the development, testing, and evaluation of the GUI-Based Expense Tracker System, the researchers concluded that the system successfully achieved its main objective of creating a desktop-based application that can help users record, organize, monitor, update, delete, and summarize expense records efficiently. The system was developed using Python as the main programming language, Tkinter for the graphical user interface, and MySQL as the database management system.

The system effectively addressed the problems commonly encountered in manual expense tracking, such as disorganized records, inaccurate calculations, missing entries, and difficulty in reviewing past transactions. Through the use of a computerized system, users can manage their expense records in a faster, more organized, and more accurate manner.

The Expense Tracker System successfully performed the essential CRUD operations, including adding, viewing, updating, and deleting expense records. It also included additional features such as category filtering, clear all records, analytics report generation, dashboard summaries, light/dark mode, keyboard shortcuts, and database connectivity. These features made the system more functional and user-friendly.

The integration of MySQL provided permanent storage for expense records. This allowed users to access saved data even after the application was closed. It also improved data organization, retrieval, and reliability. The use of a database made the system more dependable compared to manual records or temporary data storage.

The system also improved accuracy through input validation and error handling. It was able to detect empty fields, non-numeric amount inputs, invalid date formats, and actions performed without selecting a record. These validation features helped prevent incorrect or incomplete data from being saved in the database.

The graphical user interface developed using Tkinter improved user interaction and usability. Through buttons, tabs, tables, menus, dropdown lists, and dialog boxes, users can operate the system easily without using complex commands. The dashboard and report window also helped users view their total expenses, today's expenses, monthly expenses, and highest-spend category more conveniently.

The results of functional testing showed that the system performed successfully under both valid and invalid input conditions. All tested functions produced the expected results, which means that the system is functional, reliable, and stable within the scope of the study.

Overall, the GUI-Based Expense Tracker System demonstrates the practical application of object-oriented programming, graphical user interface development, database integration, input validation, and system testing. It provides a useful and organized tool for basic expense management and serves as a strong foundation for future improvements.

Recommendations

Based on the findings and limitations of the study, the following recommendations are proposed for the improvement of the Expense Tracker System:

1. Future developers should add a user login and authentication system to improve the security of expense records. This will allow only authorized users to access and manage the system.
2. The system should support multiple users so that different individuals can manage their own expense records separately. This feature would make the system more useful for families, organizations, or small businesses.
3. The graphical user interface may be improved further by making it more modern, responsive, and visually appealing. A cleaner layout and improved design can enhance user experience.

4. The system may include automatic backup and recovery features to prevent data loss in case of system errors, accidental deletion, or database problems.
5. Future versions should include export options such as PDF, Excel, or CSV reports. This will allow users to print, share, or save financial summaries more conveniently.
6. The system may be upgraded to include advanced analytics such as weekly, monthly, and yearly expense charts. These visual reports can help users better understand their spending patterns.
7. Budget-setting features may also be added so users can set spending limits for each category. The system may notify users when they are close to exceeding their budget.
8. The system should include stronger confirmation prompts for important actions, especially deleting individual records and clearing all records, to avoid accidental data loss.
9. Future researchers may consider developing a mobile or web-based version of the system so users can access their expense records from different devices.
10. The system may be improved by adding cloud storage or online database support for better accessibility, synchronization, and long-term data management.
11. Future developers may conduct usability testing with actual users to gather feedback about the system's interface, features, and ease of use. This feedback can guide further improvements.

In conclusion, while the Expense Tracker System already performs its intended functions effectively, it can still be enhanced by adding security, multi-user access, automatic backup, export options, budget monitoring, and advanced reporting features. These improvements would make the system more efficient, reliable, and suitable for broader personal, academic, and organizational use.

ACKNOWLEDGEMENT

The researchers wish to convey their deepest appreciation to Engr. Meshelle N. Fabro, PCpE, for her steadfast mentorship, expert direction, and dedicated supervision over the course of this study. Her profound insights, constructive critiques, and constant encouragement were vital to the refinement and successful execution of this research.

Heartfelt gratitude is also extended to the Eulogio "Amang" Rodriguez Institute of Science and Technology (EARIST), specifically the Department of Computer Engineering, along with its esteemed faculty. Their provision of essential academic knowledge, technical facilities, and institutional backing made the completion of this project possible. The institution's dedication to educational excellence, innovation, and scholarly research served as a cornerstone for this study's success.

Special Recognition Is Also Given to the Families and Friends of the Researchers for Their Steadfast Support, Understanding, and Patience. Their Motivating Presence Served as a Constant Source of Inspiration and Fortitude Amidst the Hurdles Faced Throughout the Research Journey.

Above All, the Researchers Express Their Utmost Gratitude to Almighty God for Granting the Wisdom, Clarity, Strength, and Resilience Required to Navigate This Endeavor. This Work Is Humbly Dedicated to His Greater Glory, in Profound Recognition That Every Milestone Achieved Was Realized Through His Grace and Blessings.

REFERENCES

1. Bhavani, V. T., Chinthana, S., Aishwarya, K., Shweta, B., & Nirmala, S. (2025). Implementation of an expense tracker using Python. *International Journal for Multidisciplinary Research*.
2. Birajdar, R. S. (2025). Personal expense tracker: Advancing financial management through software solutions. *IIP International Multidisciplinary Research Journal*.
3. Bourgeois, D. T., Smith, J. L., Wang, S., & Mortati, J. (2019). *Information systems for business and beyond*. Saylor Foundation.
4. Laudon, K. C., & Laudon, J. P. (2012). *Management information systems: Managing the digital firm* (12th ed.). Pearson.

5. MySQL. (n.d.). *MySQL documentation*. Oracle.
6. O'Brien, J. A., & Marakas, G. M. (2011). *Management information systems* (10th ed.). McGraw-Hill.
7. Python Software Foundation. (n.d.). *Python documentation*.
8. Python Software Foundation. (n.d.). *Tkinter — Python interface to Tcl/Tk*.
9. Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., & Diakopoulos, N. (2016). *Designing the user interface: Strategies for effective human-computer interaction* (6th ed.). Pearson.
10. Silberschatz, A., Korth, H. F., & Sudarshan, S. (2019). *Database system concepts* (7th ed.). McGraw-Hill Education.

About The Authors

Rhiane Biatrice P. Arceso is a dedicated student of Eulogio “Amang” Rodriguez Institute of Science and Technology who is passionate about learning and personal growth. She is known for her commitment to academic excellence, teamwork, and creativity in accomplishing various school projects and activities.

Jade Monique L. Bazarte is a passionate student of Eulogio “Amang” Rodriguez Institute of Science and Technology who demonstrates creativity, determination, and dedication in her academic pursuits. She actively participates in school activities and values continuous learning and growth.

Fresly Ann T. Cagadas is a dedicated student of Eulogio “Amang” Rodriguez Institute of Science and Technology who is eager to expand her knowledge and enhance her skills through academic learning and collaboration. She values hard work, responsibility, and teamwork in achieving both personal and academic success.

Lance Louis B. Dacara is a hardworking student from Eulogio “Amang” Rodriguez Institute of Science and Technology. He continuously develops his skills and knowledge through active participation in academic tasks and collaborative projects, showing dedication and perseverance in his studies.

Gabrielle B. Reyes is a motivated student of Eulogio “Amang” Rodriguez Institute of Science and Technology who values discipline, cooperation, and innovation. He strives to contribute meaningful ideas and perform his responsibilities effectively in both individual and group activities.

Jhanna Paula R. Sibug is a responsible and goal-oriented student from Eulogio “Amang” Rodriguez Institute of Science and Technology. She is committed to improving her academic abilities and works diligently to achieve success while maintaining positive collaboration with her peers.

Engr. Meshelle N. Fabro is a licensed Professional Computer Engineer and educator at the Eulogio “Amang” Rodriguez Institute of Science and Technology (EARIST). She is currently pursuing a Master of Science in Computer Engineering at Bulacan State University and has professional industry experience as a Technical Engineer at Hewlett-Packard (HP) and an IT Engineer at IBM. She is also a TESDA CSS NC II and Trainers Methodology Level I (TM1) holder, demonstrating her commitment to technical excellence and competency-based education. Passionate about teaching and innovation, she is dedicated to guiding Computer Engineering students in becoming industry-ready, competent, and highly skilled individuals equipped for the evolving demands of the technology industry.